Scientific & Technical Information Center
S T I C
USPTO

# STIC Search Report
EIC 2100

**STIC Database Tracking Number: 111218**

| | |
|---|---|
| **TO: Trenton Roche**<br>**Location: 5D40**<br>**Art Unit : 2124**<br>**Wednesday, December 31, 2003**<br><br>**Case Serial Number: 09/667430** | **From: David Holloway**<br>**Location: EIC 2100**<br>**PK2-4B30**<br>**Phone: 308-7794**<br><br>**david.holloway@uspto.gov** |

## Search Notes

Dear Examiner Roche,

Attached please find your search results for above-referenced case.
Please contact me if you have any questions or would like a re-focused search.

David

```
Set     Items   Description
S1      12123   (MUTABIL? OR CHANG?(2N)INITIAL?)
S2        259   S1(4N)(VARIABL? OR OBJECT? OR CLASS? OR FIELD?)
S3    8236507   DETECT? OR IDENTIF? OR ID OR FIND? OR LOCAT? OR CLASSIF?
S4         76   S2 AND S3
S5         55   RD (unique items)
S6         49   S5 NOT PY>2000
File   8:Ei Compendex(R) 1970-2003/Dec W3
         (c) 2003 Elsevier Eng.  Info. Inc.
File  35:Dissertation Abs Online 1861-2003/Nov
         (c) 2003 ProQuest Info&Learning
File  65:Inside Conferences 1993-2003/Dec W4
         (c) 2003 BLDSC all rts. reserv.
File   2:INSPEC 1969-2003/Dec W2
         (c) 2003 Institution of Electrical Engineers
File  94:JICST-EPlus 1985-2003/Dec W3
         (c)2003 Japan Science and Tech Corp(JST)
File 111:TGG Natl.Newspaper Index(SM) 1979-2003/Dec 31
         (c) 2003 The Gale Group
File 233:Internet & Personal Comp. Abs. 1981-2003/Sep
         (c) 2003 EBSCO Pub.
File 144:Pascal 1973-2003/Dec W2
         (c) 2003 INIST/CNRS
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 1998 Inst for Sci Info
File  34:SciSearch(R) Cited Ref Sci 1990-2003/Dec W4
         (c) 2003 Inst for Sci Info
File  62:SPIN(R) 1975-2003/Nov W2
         (c) 2003 American Institute of Physics
File  99:Wilson Appl. Sci & Tech Abs 1983-2003/Nov
         (c) 2003 The HW Wilson Co.
File  95:TEME-Technology & Management 1989-2003/Dec W2
         (c) 2003 FIZ TECHNIK
```

**6/5/14 (Item 6 from file: 35)**
DIALOG(R)File 35:Dissertation Abs Online

01427071 ORDER NO: AADAA-I0576116

**FUNCTIONAL ENCAPSULATION AND TYPE RECONSTRUCTION IN A STRONGLY-TYPED, POLYMORPHIC LANGUAGE**

Author: GUPTA, SHAIL ADITYA
Degree: PH.D.
Year: 1995
Corporate Source/Institution: MASSACHUSETTS INSTITUTE OF TECHNOLOGY ( 0753)
Chairman: FREDERIC R. MORGENTHALER
Source: VOLUME 56/04-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 2133.
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

Static type systems are traditionally used to prevent run-time type-errors in user programs and to assign appropriate storage representations to objects during compilation. In this thesis, we explore some new ways of using static type information in the design, compilation, and execution of programs written in a strongly-typed, polymorphic language.

Programmers often **find** it useful to know whether or not a particular data-structure may be updated outside a given control block. Information about an **object** 's non- **mutability** helps compiler optimizations, improves aliasing and dependence analyses, and permits unrestricted caching of functional data at run-time. In the first part of this thesis, we present a safe, static mechanism for functional encapsulation of imperative data-structures using a powerful type system based on closure types and regions. We introduce a new language construct called close which delimits the scope of side-effects on imperative objects and converts them into functional objects outside that scope. This mechanism may be used to build efficient, high-level, functional data-abstractions within a language using its low-level, imperative constructs. Type-safety and non- **mutability** of closed **objects** is guaranteed by a semantic soundness theorem that ensures consistency between the static and the dynamic semantics. The type system is presented in the context of **Id** , which is a strongly-typed, polymorphic, higher-order language, and it easily simplifies to a first-order, monomorphic language such as C or Fortran.

In the second part of the thesis, we develop a general, compiler-directed methodology for complete type reconstruction of run-time objects in a polymorphic language without using any run-time type-tags. Run-time type reconstruction is carried out by instantiating static type information for each function activation frame present within the dynamic call tree. Additional type-hints are inserted automatically at compile-time and are decoded at run-time to ensure complete type reconstruction. We present the necessary compiler analysis and the type reconstruction algorithm and prove their correctness. This technique has been used successfully for displaying run-time objects within the **Id** source debugger for Monsoon and to perform tagless garbage collection in the *T architecture. We describe the latter application in detail, comparing its performance with other schemes for automatic storage reclamation. (Copies available exclusively from MIT Libraries, Rm. 14-0551, Cambridge, MA 02139-4307. Ph. 617-253-5668; Fax 617-253-1690.)

**6/5/25 (Item 4 from file: 2)**
DIALOG(R)File 2:INSPEC

4413223 INSPEC Abstract Number: C9307-4240-010
**Title: On specialization hierarchies of mutable objects**
Author(s): Maung, I.
Author Affiliation: Dept of Comput., Brighton Univ., UK
Journal: Bulletin of the European Association for Theoretical Computer Science no.49 p.165-74
Publication Date: Feb. 1993 Country of Publication: Netherlands
CODEN: BEASDU ISSN: 0252-9742
Language: English Document Type: Journal Paper (JP)
Treatment: Theoretical (T)
Abstract: The author presents an elementary mathematical formulation of the **classification** hierarchies of object-orientation, in particular hierarchies in which subclasses are specializations of their superclasses. All the results presented are essentially trivial and almost surely unoriginal. However, the author believes that this is one of the first serious applications of lattice theory to OO **classification** hierarchies. It also provides us with some original insights-the author shows that the possible specialization hierarchies of a **class** are limited by the **mutability** of its instances, and gives precise conditions under which (1) a class must be concrete, and (2) it can be made abstract. The author also applies the theory to the design of OO class hierarchies, and to the formalization of the fine-grained hierarchies of Johnson and Rees (GEC-Marconi Res. Tech. Rep. Y/240/1809, 1991). (9 Refs)
Subfile: C
Descriptors: inheritance; object-oriented methods
Identifiers: class instances; specialization hierarchies; mutable objects ; object-orientation; subclasses; superclasses; lattice theory; OO **classification** hierarchies; mutability; OO class hierarchies; fine-grained hierarchies
Class Codes: C4240 (Programming and algorithm theory); C6110J ( Object-oriented programming)

Scientific & Technical Information Center
S T I C
USPTO

# STIC EIC 2100 Search Request Form

111218
123

Today's Date: 12/31/03

What date would you like to use to limit the search?
Priority Date: 09/21/2000 Other:

Name Trent Roche

AU 2124 Examiner # 79908

Room # 5D40 Phone 305-4627

Serial # 09/667,430

Format for Search Results (Circle One):
[✓ PAPER] DISK EMAIL

Where have you searched so far?
[✓ USP] [✓ DWPI] [✓ EPO] [✓ JPO] [✓ ACM] [✓ IBM TDB]
[✓ IEEE] INSPEC SPI Other Citeseer

Is this a "Fast & Focused" Search Request? (Circle One) [✓ YES] NO
A "Fast & Focused" Search is completed in 2-3 hours (maximum). The search must be on a very specific topic and meet certain criteria. The criteria are posted in EIC2100 and on the EIC2100 NPL Web Page at http://ptoweb/patents/stic/stic-tc2100.htm.

What is the topic, novelty, motivation, utility, or other specific details defining the desired focus of this search? Please include the concepts, synonyms, keywords, acronyms, definitions, strategies, and anything else that helps to describe the topic. Please attach a copy of the abstract, background, brief summary, pertinent claims and any citations of relevant art you have found.

This is a method of detecting the mutability of variables, objects, fields or classes in Java. A variable is termed "mutable" if it's state ever changes after initialization.

STIC Searcher David Holloway Phone 308-7794

Date picked up 12-31-03 Date Completed 12-31-03

sira
Search and Information Resources Administration

```
Set     Items   Description
S1          2   MUTABILITY(2N)OBJECT?(10N)(DETECT? OR IDENTIF?)
File 349:PCT FULLTEXT 1979-2002/UB=20031225,UT=20031218
         (c) 2003 WIPO/Univentio
File 351:Derwent WPI 1963-2003/UD,UM &UP=200382
         (c) 2003  Thomson Derwent
```

```
Set     Items   Description
S1      16358   (MUTABIL? OR CHANG?(2N)INITIAL?)
S2        269   S1(4N)(VARIABL? OR OBJECT? OR CLASS? OR FIELD?)
S3   13433384   DETECT? OR IDENTIF? OR ID OR FIND? OR LOCAT? OR CLASSIF?
S4         14   S2 (10N) S3
S5         10   RD (unique items)
S6          9   S5 NOT PY>2000
File 275:Gale Group Computer DB(TM) 1983-2003/Dec 31
         (c) 2003 The Gale Group
File  47:Gale Group Magazine DB(TM) 1959-2003/Dec 25
         (c) 2003 The Gale group
File 636:Gale Group Newsletter DB(TM) 1987-2003/Dec 31
         (c) 2003 The Gale Group
File  16:Gale Group PROMT(R) 1990-2003/Dec 31
         (c) 2003 The Gale Group
File 624:McGraw-Hill Publications 1985-2003/Dec 30
         (c) 2003 McGraw-Hill Co. Inc
File 484:Periodical Abs Plustext 1986-2003/Dec W2
         (c) 2003 ProQuest
File 613:PR Newswire 1999-2003/Dec 31
         (c) 2003 PR Newswire Association Inc
File 813:PR Newswire 1987-1999/Apr 30
         (c) 1999 PR Newswire Association Inc
File 141:Readers Guide 1983-2003/Nov
         (c) 2003 The HW Wilson Co
File 621:Gale Group New Prod.Annou.(R) 1985-2003/Dec 26
         (c) 2003 The Gale Group
File 674:Computer News Fulltext 1989-2003/Dec W3
         (c) 2003 IDG Communications
File  88:Gale Group Business A.R.T.S. 1976-2003/Jan 02
         (c) 2003 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 635:Business Dateline(R) 1985-2003/Dec 31
         (c) 2003 ProQuest Info&Learning
File  15:ABI/Inform(R) 1971-2003/Dec 31
         (c) 2003 ProQuest Info&Learning
File   9:Business & Industry(R) Jul/1994-2003/Dec 29
         (c) 2003 Resp. DB Svcs.
File  13:BAMP 2003/Dec W3
         (c) 2003 Resp. DB Svcs.
File 810:Business Wire 1986-1999/Feb 28
         (c) 1999 Business Wire
File 610:Business Wire 1999-2003/Dec 31
         (c) 2003 Business Wire.
File 647:CMP  Computer Fulltext 1988-2003/Dec W3
         (c) 2003 CMP Media, LLC
File 148:Gale Group Trade & Industry DB 1976-2003/Dec 26
         (c)2003 The Gale Group
File 634:San Jose Mercury  Jun 1985-2003/Dec 29
         (c) 2003 San Jose Mercury News
```

PriorArt database

**pri rartdatabase.com**
http://www.priorartdatabase.com/

**Searching:** mutability
Search results are limited to a maximum of 25 records on this site. Additionally, no records within the last 60 days are returned in search results.

For more comprehensive searching that is not limited to 25 records and can search even the most recent documents, consider purchasing a pass to the Prior Art Database Premium service.

**IP Authentication Header (RFC2402)** [000002977]
2000-09-13
The IP Authentication Header (AH) is used to provide connectionless integrity and data origin ...

**Method for detecting caching opportunities in software.** [000011999]
2003-04-01
This article describes heuristics for the identification of caching opportunities in software. ...

**A URN Namespace for IETF Documents (RFC2648)** [000003236]
2000-09-13
A system for Uniform Resource Names (URNs) must be capable of supporting new naming systems. As an ...

**An Expedited Forwarding PHB (Per-Hop Behavior) (RFC3246)** [000007370]
2002-03-20
This document defines a PHB (per-hop behavior) called Expedited Forwarding (EF). The PHB is a ...

**Message Context for Internet Mail (RFC3458)** [000011353]
2003-02-14
This memo describes a new RFC 2822 message header, "Message-Context". This header provides ...

**A Delay Bound alternative revision of RFC 2598 (RFC3248)** [000007371]
2002-03-20
For historical interest, this document captures the EF Design Team's proposed solution, preferred ...

**An Expedited Forwarding PHB (RFC2598)** [000003185]
2000-09-13
The definition of PHBs (per-hop forwarding behaviors) is a critical part of the work of the ...

**Thermophilic DNA polymerase** [000001511]
2000-09-12
The invention relates to a substantially pure thermostable DNA polymerase. Preferably, the DNA ...

```
Set      Items   Descript[illegible]n
S1        2441   (MUTABIL? OR CHANG?(2N)INITIAL?)
S2          16   S1(2N)(VARIABL? OR OBJECT? OR CLASS?)
S3     2983641   DETECT? OR IDENTIF? OR ID OR FIND? OR LOCAT? OR CLASSIF?
S4           3   S2 AND S3
S5         602   S1 AND S3
S6         107   S5 AND (VARIAB? OR OBJECT? OR FIELD? OR CLASS?)
S7          52   S6 AND (JAVA OR OBJECT()ORIENT? OR OO OR C OR C? ? OR C?? ?
               OR SMALLTALK?)
S8           4   "C+" OR "C++"
S9           0   S6 AND S8
S10         44   S7 NOT AD>20000921
S11          9   S10 AND IC=G06F?
S12         25   S6 AND (JAVA OR OBJECT()ORIENT? OR OO OR C OR SMALLTALK?)
S13          8   S12 AND IC=(G06F? OR H04L?)
S14          0   S13 NOT (S3 OR S11)
S15         13   S2 NOT (S4 OR S11)
File 347:JAPIO Oct 1976-2003/Aug(Updated 031202)
         (c) 2003 JPO & JAPIO
File 350:Derwent WPIX 1963-2003/UD,UM &UP=200382
         (c) 2003  Thomson Derwent
```

**15/5/2 (Item 2 from file: 347)**

03830356 **Image available**
METHOD AND DEVICE FOR NONLINEAR OPTIMIZATION

PUB. NO.: 04-195456 [JP 4195456 A]
PUBLISHED: July 15, 1992 (19920715)
INVENTOR(s): TAMURA MASAYOSHI
NISHIKATA MASAHARU
APPLICANT(s): HITACHI LTD [000510] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 02-322957 [JP 90322957]
FILED: November 28, 1990 (19901128)
INTL CLASS: [5] G06F-015/20; G06F-015/31
JAPIO CLASS: 45.4 (INFORMATION PROCESSING -- Computer Applications)
JOURNAL: Section: P, Section No. 1446, Vol. 16, No. 525, Pg. 84, October 28, 1992 (19921028)

ABSTRACT

PURPOSE: To avoid an optimum variable from being obtained again by performing nonlinear optimization in the case of a sufficient difference between the direction in which the **variable** is **changed** from an **initial variable** and directions to all known and locally optimum variables.

CONSTITUTION: Information and processing procedures inputted from an input device 1101 are stored in a storage device 1102. The processing result or the like is outputted to an output device 1103. In this case, nonlinear optimization is performed only when there is a high probability that an unknown and locally optimum variable can be found from the initial variable generated for a nonlinear optimization problem having plural locally optimum variables, but nonlinear optimization is not performed when there is a high probability that known and locally optimum variables are obtained again. Thus, unnecessary nonlinear optimization is not performed to obtain plural locally optimum variables

```
 4/5/2     (Item 2 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2003  Thomson Derwent. All rts. reserv.

014595604     **Image available**
WPI Acc No: 2002-416308/200244
XRPX Acc No: N02-327577
   Detecting  mutability of program component variables and objects by
  initializing class and instance variables on completion of corresponding
  method
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC  ); IBM UK LTD (IBMC  )
Inventor: BIBERSTEIN M; KOVED L; MENDELSON B; PORAT S
Number of Countries: 096  Number of Patents: 002
Patent Family:
Patent No      Kind   Date     Applicat No     Kind   Date      Week
WO 200225425    A2  20020328  WO 2001GB4158     A   20010917  200244  B
AU 200187880    A   20020402  AU 200187880      A   20010917  200252

Priority Applications (No Type Date): US 2000667430 A 20000921
Patent Details:
Patent No  Kind Lan Pg   Main IPC    Filing Notes
WO 200225425  A2 E  60 G06F-009/00
   Designated States (National): AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA
   CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN
   IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ
   PH PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW
   Designated States (Regional): AT BE CH CY DE DK EA ES FI FR GB GH GM GR
   IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TR TZ UG ZW
AU 200187880  A        G06F-009/00   Based on patent WO 200225425

Abstract (Basic): WO 200225425 A2
        NOVELTY - Method is for a Java environment and consists in
    determining whether a variable could be state modified and performing
    encapsulation analysis to see if it could undergo a second type of
    state modification outside the program component. A variable or object
    is mutable if its state ever changes after it is initialized, a field
    is mutable if any corresponding variable is, and a class is mutable if
    any instance fields implemented by it are. Possible breakage of
    variable encapsulation if a method within the program component causes
    a mutable object reachable from the variable to become accessible to
    methods not within the component is  detected .
        DETAILED DESCRIPTION - Class variables are initialized on
    completion of the corresponding clinit method, and instance variables
    are initialized on completion of the corresponding init method. There
    are INDEPENDENT CLAIMS for (1) a device for  detecting   mutability  of
     variables ,  objects , fields and classes in a program component
    written in an object-oriented programming language, (2) a computer
    system for  detecting   mutability  of  variables ,  objects , fields
    and classes in a program component written in an object-oriented
    programming language.
        USE - Method is for  detecting   mutability  of  variables ,
    objects , fields and classes in an object-oriented programming language
    component.
        ADVANTAGE - Method is for  identifying  and stopping isolation
    faults.
        DESCRIPTION OF DRAWING(S) - The figure shows a block diagram of a
    mutability analyzer.
        pp; 60 DwgNo 1/4
Title Terms:  DETECT ; PROGRAM; COMPONENT; VARIABLE; OBJECT; INITIALISE;
  CLASS; INSTANCE; VARIABLE; COMPLETE; CORRESPOND; METHOD
Derwent Class: T01
International Patent Class (Main): G06F-009/00
File Segment: EPI
```